REMARKS

This paper is being provided in response to the July 10, 2007 Final Office Action for the above-referenced U.S. patent application. In this response, Applicants have amended claims 1 and 10 in order to clarify that which Applicants deem to be the invention. Applicants respectfully submit that the amendments to the claims are supported by the originally filed application.

The rejection of claims 1, 6, 10, and 15 under 35 U.S.C. 103 (a) as being unpatentable over U. S. patent number 5,901,327 to Ofek (hereinafter "Ofek") in view of U.S. patent number 6,038,651 to VanHuben, et al. (hereinafter "VanHuben") and U.S. patent application number 2002/0116404 to Cha et al. (hereinafter "Cha") is hereby traversed and reconsideration thereof is respectfully requested.

Claim 1, as amended herein, is for a method of using a local storage device to read desired data while the data is being transferred from the local storage device to a remote storage device. The method includes, if the desired data is entirely in a cache of the local storage device, the local storage device returning the data from the cache. The cache of the local storage device is recited as containing data written to the local storage device begun after a first time and before a second time that is associated with a first chunk of data and containing data written to the local storage device begun after the second time that is associated with a second chunk of data different from the first chunk of data. After completion of all writes associated with the first chunk of data, the local storage device initiates transfer of writes associated with the first chunk

of data to the remote storage device, an order of the transfer from the local storage device to the remote storage device of data from the first chunk being independent of an order in which data writes of the first chunk are provided to the local storage device. If the desired data is not entirely in the cache of the local storage device, data is read from the remote storage device to the local storage device and the local storage device merges the data from the remote storage device with data from the cache of the local storage device at the local storage device. Claim 6 depends from claim 1.

Claim 10, as amended herein, is for computer software, stored in a computer-readable medium, that reads desired data while the data is being transferred from the local storage device to a remote storage device. The software includes executable code that returns the data from the cache if the desired data is entirely in a cache of the local storage device. The cache of the local storage device contains data written to the local storage device begun after a first time and before a second time that is associated with a first chunk of data and contains data written to the local storage device begun after the second time that is associated with a second chunk of data different from the first chunk of data. After completion of all writes associated with the first chunk of data, the local storage device initiates transfer of writes associated with the first chunk of data to the remote storage device, an order of the transfer from the local storage device to the remote storage device of data from the first chunk being independent of an order in which data writes of the first chunk are provided to the local storage device. The software also includes executable code that reads data from the remote storage device to the local storage device and merges the data from the remote storage device with data from the cache of the local storage device at the local storage device if the desired data is not entirely in the cache of the local

storage device. Claim 15 depends from claim 10.

The present claimed invention reads data from a remote storage device while the data is being transferred from a local storage device to the remote storage device by first checking if the data is in the cache of the local storage device. If the data is available in the cache of the local storage device, the data being read from the remote storage device is obtained from the cache of the local storage device. The cache of the local storage device contains data written to the local storage device begun after a first time and before a second time that is associated with a first chunk of data and contains data written to the local storage device begun after the second time that is associated with a second chunk of data different from the first chunk of data. After completion of all writes associated with the first chunk of data, the local storage device initiates transfer of writes associated with the first chunk of data to the remote storage device, an order of the transfer from the local storage device to the remote storage device of data from the first chunk being independent of an order in which data writes of the first chunk are provided to the local storage device. Thus, the present claimed invention reads data being stored on the remote storage device without having to always obtain the data from the remote storage device even though the transfer mechanism between the local and remote storage devices is relatively complex.

Ofek discloses a system and method for automatically providing and maintaining a copy or mirror of data stored at a location remote from the main or primary storage device. Data is retrieved from a remote device through a host data processing system. The host 12 writes data to and reads data from the primary data storage system 14. The host central processing unit 212

can also be provided with host remote mirroring software 213 so that the data processing system can be configured and monitored from a user interface of the host central processing unit. Host application programs can also interface with the remote mirroring facility of the data storage systems 214, 246 via the host remote mirroring software 213. During a read access, the channel adapter accesses the cache. If the data requested by the host is not found in the cache, the data is fetched by a disk adapter from the disk storage in the data storage system and loaded into the cache. Column 14 beginning at line 43 of Ofek discloses that, under the abnormal condition of the data being entirely absent from the data storage system due to a disk drive failure, requests for data access to a primary volume (R1) can be satisfied by obtaining the requested data from the secondary volume (R2) in the remote data store system.

Ofek also discloses use of a time stamp or sequence number in connection with data transmitted from the primary volume to the secondary volume. Column 38, lines 43-57 of Ofek provide:

> The time stamp or sequence number can be used by the remote data storage system to *detect link transmission problems* and to write to its cache in proper sequence data from commands received from various links and link adapters despite possible delay of some commands due to link failure. In an alternative arrangement, each link queue entry or corresponding link buffer entry could be marked with a time stamp or sequence number at the time the link queue entry is inserted at the tail of the link queue, so that step 544 could be eliminated. Moreover, in the short distance option configuration having a single link, *time stamps or sequence numbers would not be needed, because each command could be transmitted over the link, received, and acknowledged before the next command in the link queue would be transmitted.* (emphasis added)

VanHuben discloses a remote resource management system for managing resources in a symmetrical multiprocessing comprising a plurality of clusters of symmetric multiprocessors

having interfaces between cluster nodes of the symmetric multiprocessor system. Each cluster of the system has a local interface and interface controller. There are one or more remote storage controllers each having its local interface controller, and a local-to-remote data bus. The remote resource manager manages the interface between two clusters of symmetric multiprocessors each of which clusters has a plurality of processors, a shared cache memory, a plurality of I/O adapters and a main memory accessible from the cluster.

Cha discloses a differential logging scheme for recovering from a failure in a transaction system that allows commutative and associative recovery operations. Cha discloses taking a before-image of a primary database in main memory before an update to the primary database, taking an after-image of the primary database after the update, and generating a differential log by applying bit-wise exclusive-OR (XOR) between the before-image and the after-image. Either a redo or undo operation may be performed by applying XOR between the log and the before-image or after-image. Multiple logs may be generated in this way. Since XOR operations are commutative and associative, correct recovery is possible regardless of the creation sequence and/or application sequence of log records.

Applicants respectfully submit that neither Ofek, nor VanHuben, nor Cha nor any combination thereof show, teach, or suggest features recited in independent claims 1 and 10, including the feature of the cache of the local storage device containing data written to the local storage device begun after a first time and before a second time that is associated with a first chunk of data and containing data written to the local storage device begun after the second time that is associated with a second chunk of data different from the first chunk of data where, after

completion of all writes associated with the first chunk of data, the local storage device initiates transfer of writes associated with the first chunk of data to the remote storage device, an order of the transfer from the local storage device to the remote storage device of data from the first chunk being independent of an order in which data writes of the first chunk are provided to the local storage device. The transfer mechanism for each of Ofek and VanHuben appears to be much simpler and apparently involves transferring data as it is written. In fact, Ofek specifically discloses time stamping or sequence number stamping data that is transferred and that it is an error if data is received out of sequence (see, for example, column 38, lines 43-57 of Ofek). The other mechanism Ofek discloses is doing away with sequence numbers or time stamps altogether (in short distance configurations) by requiring that each transmission be acknowledged as being received before making a subsequent transmission.

The Office Action apparently relies on Cha to overcome the above-noted deficiencies of Ofek and VanHuben. However, Cha does not teach an order of the transfer from the local storage device to the remote storage device of data from the first chunk being independent of an order in which data writes of the first chunk are provided to the local storage device as recited in Applicants' independent claims. Instead, Cha discloses order independence of the application of special *log files* to redo or undo changes to a database. The log files of Cha do not contain data from any version of the database and, in fact, Cha specifically teaches that they should not contain data for his system to work. See, for example, Cha's figure 6B (contains data from database) and figure 6C (does not contain data from database) and the corresponding discussion. Furthermore, the order independent application of the log files of Cha to the data does not result in any intermediate versions of the data, but only provides a correct final version of the data.

Said differently, order independent application of the log files only provides a correct final version of the data. For example, the right-hand portion of figure 6E shows intermediate results of 15, 8, 9, 1, and 6 that do not correspond to any intermediate version of the illustrated data at all.

In addition, neither Ofek, nor VanHuben discuss the use of log files. Thus, it is not clear how one would adapt either or both to use the log files of Cha - especially since Cha discloses a single system that uses log files. Thus, for example, for Ofek, would the log files be provided at the primary (R1) site, the secondary (R2) site, or both sites and which entity(s) would be responsible for gathering accurate log information and transmitting that information between the sites? What entities of Ofek would be responsible for the integrity of log information - especially after a recovery operation? Even if the references are combinable for reasons set forth in the Office Action, there's nothing in any of the references that would teach one of ordinary skill in the art to adapt the Cha reference to the systems of Okek and/or VanHuben.

Furthermore, even if one were to combine Ofek, VanHuben, and Cha as suggested by the Office Action, the resulting system would still not provide an order of the transfer from the local storage device to the remote storage device of data from the first chunk being independent of an order in which data writes of the first chunk are provided to the local storage device as recited in Applicants' independent claims. As indicated in the Office Action, neither Ofek nor VanHuben teach this and Cha only teaches this for log files, not for the data itself.

Accordingly, for reasons set forth above, Applicants respectfully request that this rejection be withdrawn.

The rejection of claims 2, 7-9, 11, and 16-18 under 35 U.S.C 103(a) as being unpatentable over Ofek, VanHuben and Cha in view of U.S. patent number 6,880,045 to Pong et al. (hereinafter "Pong") is hereby traversed and reconsideration thereof is respectfully requested.

Claims 2 and 7-9 depend from claim 1, discussed above. Claims 11 and 16-18 depend from claim 10, discussed above.

The Ofek, VanHuben and Cha references are discussed above.

As set forth in the Office Action, Pong teaches prior to reading data from the remote storage device to the local storage device, creating a temporary storage device at the local storage device if there is data from the local storage device that is to be read.

Applicant respectfully submits that the deficiencies of Ofek, VanHuben, and Cha with respect to claims 1 and 10, discussed above, are not overcome by the addition of the Pong reference. Accordingly, Applicant respectfully requests that this rejection be withdrawn.

The rejection of claims 3-5, and 12-14 under 35 U.S.C 103(a) as being unpatentable over Ofek, VanHuben, Cha and Pong and further in view of U.S. patent number 6,012,063 Bodnar et

al. (hereinafter "Bodnar") is hereby traversed and reconsideration thereof is respectfully requested.

Claims 3-5 depend from claim 1, discussed above. Claims 12-14 depend from claim 10, discussed above.

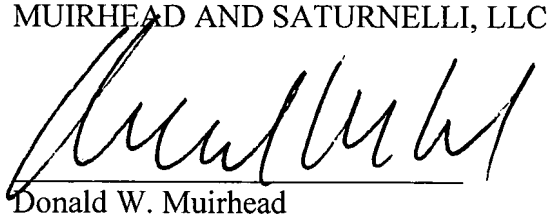
The Ofek, VanHuben, Cha and Pong references are discussed above.

As set forth in the Office Action, Bodnar teaches having a temporary storage area that is a scratch slot.

Applicant respectfully submits that the deficiencies of Ofek, VanHuben and Cha (and Pong) with respect to claims 1 and 10, discussed above, are not overcome by the addition of the Bodnar reference. Accordingly, Applicant respectfully requests that this rejection be withdrawn.

Based on the above, Applicants respectfully request that the Examiner reconsider and withdraw all outstanding rejections and objections. Favorable consideration and allowance are earnestly solicited. Should there be any questions after reviewing this paper, the Examiner is invited to contact the undersigned at 508-898-8603.

Respectfully submitted,
MUIRHEAD AND SATURNELLI, LLC



October 9, 2007
Date

Donald W. Muirhead
Registration No. 33,978

Muirhead and Saturnelli, LLC
200 Friberg Parkway, Suite 1001
Westborough, MA 01581
508-898-8601 (main)
508-898-8602 (fax)

Customer No. 52427